This depth information can help determine whether a particular primitive, or part of the primitive, is blocked from view by another primitive. Referring again to **FIGS. 5A and 5B**, depth may increase going "into" the 3D space from the virtual display **144**. Thus, object **150** may be at a greater depth than object **140**. Similarly, the depth information associated with projected polygon **152** may indicate a greater depth than that of polygons **146** and **148**. Therefore, if object **140** is opaque, polygon **152** is hidden and may not be displayed on the display unit.

[**0075**] The data relating to a 2D primitive may also include transparency information, such as a value that indicates the degree of transparency of the primitive. In some typical 3D graphic systems, a primitive's "alpha value" indicates the degree of the primitive's transparency. Frequently, the alpha value may be between the values 0 and 1, inclusive, where, for example, a 0 indicates the primitive is completely transparent and a 1 indicates the primitive is completely opaque. In other 3D graphics systems, the transparency information may merely include a flag that indicates whether or not the primitive is transparent, and the degree of transparency is the same for all transparent primitives. For example, the degree of transparency may be fixed at 50% for transparent objects. In 3D graphics systems that employ "alpha values," a technique often referred to as "alpha blending" may be used to generate a display in which an object appears to be transparent when viewed on the display unit.

[**0076**] Referring now to **FIGS. 3, 6A**, and **6B**, two examples of graphics processors that may be used in the gaming unit **20** are illustrated. **FIG. 6A** illustrates a graphics processor **107A** configured to receive information relating to 2D primitives from, for example, microprocessor **104**. The graphics processor **107A** may generate one or more control signals for driving display unit **70**. With the graphics processor **107A**, the transform and lighting step may be implemented, for example, with the microprocessor **104**. Optionally, graphics processors **107A** and **107B** may be configured to receive an overlay input. The overlay input may be used to provide an image that is to overlay a base image. Additionally, graphics processors **107A** and **107B** may be configured to optionally overlay the image such that it appears transparent.

[**0077**] **FIG. 6B** illustrates a graphics processor **107B** configured to receive information that may include information relating to 3D primitives, point of view, and lighting (if any). This information may be received, for example, from microprocessor **104**. In this example, the graphics processor **107B** may implement the "Transform and Lighting" processing step described above. It is to be understood that 3D graphics processing implementation can be partitioned between the graphics processor **107** and the microprocessor **104** in any number of ways. For example, much of the processing typically implemented by commercially available graphics processors could be implemented by the microprocessor **104**, thus eliminating or reducing the cost of graphics processor **107**.

[**0078**] Details of 3D graphical techniques that may be used are described in "OpenGL Reference Manual: The Official Reference Document to Open GL, Version 1.2," 3rd edition, Dave Shreiner (editor), OpenGL Architecture Review Board, Addison-Wesley Publishing, Co., 1999, ISBN: 0201657651 and "OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2," 3rd edition, Mason Woo et al. (editors), OpenGL Architecture Review Board, Addison-Wesley Publishing Co., 1999, ISBN: 0201604582, which are hereby incorporated by reference in their entirety for all purposes.

[**0079**] Additional detail pertinent to 3D graphics is available in commonly assigned U.S. patent application Ser. No. 09/927,901 (Client Reference No. P-557), entitled "Virtual Cameras and 3-D Gaming Environments in a Gaming Machine," filed Aug. 9, 2001, which is hereby incorporated by reference in its entirety for all purposes.

Overall Operation of Gaming Unit

[**0080**] One manner in which one or more of the gaming units **20** (and one or more of the gaming units **30**) may operate is described below in connection with a number of flowcharts which represent a number of portions or routines of one or more computer programs, which may be stored in one or more of the memories of the controller **100**. The computer program(s) or portions thereof may be stored remotely, outside of the gaming unit **20**, and may control the operation of the gaming unit **20** from a remote location. Such remote control may be facilitated with the use of a wireless connection, or by an Internet interface that connects the gaming unit **20** with a remote computer (such as one of the network computers **22, 32**) having a memory in which the computer program portions are stored. The computer program portions may be written in any high level language such as C, C+, C++ or the like or any low-level, assembly or machine language. By storing the computer program portions therein, various portions of the memories **102, 106** are physically and/or structurally configured in accordance with computer program instructions.

[**0081**] **FIG. 7** is a flowchart of a main operating routine **200** that may be stored in the memory of the controller **100**. Referring to **FIG. 7**, the main routine **200** may begin operation at block **202** during which an attraction sequence may be performed in an attempt to induce a potential player in a casino to play the gaming unit **20**. The attraction sequence may be performed by displaying one or more video images on the display unit **70** and/or causing one or more sound segments, such as voice or music, to be generated via the speakers **62**. The attraction sequence may include a scrolling list of games that may be played on the gaming unit **20** and/or video images of various games being played, such as video poker, video blackjack, video slots, video keno, video bingo, etc.

[**0082**] During performance of the attraction sequence, if a potential player makes any input to the gaming unit **20** as determined at block **204**, the attraction sequence may be terminated and a game-selection display may be generated on the display unit **70** at block **206** to allow the player to select a game available on the gaming unit **20**. The gaming unit **20** may detect an input at block **204** in various ways. For example, the gaming unit **20** could detect if the player presses any button on the gaming unit **20**; the gaming unit **20** could determine if the player deposited one or more coins into the gaming unit **20**; the gaming unit **20** could determine if player deposited paper currency into the gaming unit; etc.

[**0083**] The game-selection display generated at block **206** may include, for example, a list of video games that may be